Advanced Algorithms
Homework 1: Preliminaries, Divide and Conquer
CS 611, Spring 2020

Name: Kunjal Panchal
Student ID: 32126469
Email: kpanchal@umass.edu

February 2, 2020

Contents

1 Extra Credit Problem : Recurrence **3**

1.1 (a) $\mathbf{T(n) = 2T(n/2) + n^4}$ 3

1.2 (b) $\mathbf{T(n) = T(7n/10) + n}$ 3

1.3 (c) $\mathbf{T(n) = 16T(n/4) + n^2}$ 4

1.4 (d)$\mathbf{T(n) = 7T(n/3) + n^2}$ 4

1.5 (e) $\mathbf{T(n) = 7T(n/2) + n^2}$ 4

1.6 (f) $\mathbf{T(n) = 2T(n/4) + \sqrt{n}}$ 4

1.7 (g) $\mathbf{T(n) = T(n - 2) + n^2}$ 5

1 Extra Credit Problem : Recurrence

The general form for the recurrence relation for the divide and conquer algorithmic paradigm is:

$$T(n) = a \cdot T(\frac{n}{b}) + \Theta(n^\alpha) \qquad \text{for } n > 1$$
$$T(1) = \Theta(1) \qquad \text{for } n = 1$$

Let's recall the master theorem:
If we let $\beta = \log_b^a$, then the solution to the recurrence is dependent upon the relationship of α to β:

$$T(n) = \begin{cases} \Theta(n^\alpha) & \text{if } \alpha > \beta \\ \Theta(n^\beta) & \text{if } \alpha < \beta \\ \Theta(n^\alpha \log n) & \text{if } \alpha = \beta \end{cases}$$

Now, we are ready to solve the runtime of the recurrences.

1.1 (a) $\mathbf{T(n) = 2T(n/2) + n^4}$

$$a = 2, b = 2 \implies \beta = \log_2 2 = n; \alpha = 4$$

Here, $\alpha > \beta$, thus:

$$T(n) = \Theta(n^\alpha) = \Theta(n^4)$$
$$\boxed{T(n) = \Theta(n^4)}$$

1.2 (b) $\mathbf{T(n) = T(7n/10) + n}$

$$a = 1, b = 10/7 \implies \beta = \log_{10/7} 1 = 0; \alpha = 1$$

Here, $\alpha > \beta$, thus:

$$T(n) = \Theta(n^\alpha) = \Theta(n^1)$$
$$\boxed{T(n) = \Theta(n)}$$

1.3 (c) $\mathbf{T(n) = 16T(n/4) + n^2}$

$$a = 16, b = 4 \implies \beta = \log_4 16 = 2\log_4 4 = 2; \alpha = 2$$

Here, $\alpha = \beta$, thus:

$$T(n) = \Theta(n^\alpha \log n) = \Theta(n^2 \log n)$$

$$\boxed{T(n) = \Theta(n^2 \log n)}$$

1.4 (d)$\mathbf{T(n) = 7T(n/3) + n^2}$

$$a = 7, b = 3 \implies \beta = \log_3 7 = \log_7 / \log_3 = 1.72; \alpha = 2$$

Here, $\alpha > \beta$, thus:

$$T(n) = \Theta(n^\alpha) = \Theta(n^2)$$

$$\boxed{T(n) = \Theta(n^2)}$$

1.5 (e) $\mathbf{T(n) = 7T(n/2) + n^2}$

$$a = 7, b = 2 \implies \beta = \log_2 7 = \log_7 / \log_2 = 2.807; \alpha = 2$$

Here, $\alpha < \beta$, thus:

$$T(n) = \Theta(n^\beta) = \Theta(n^{\lg 7}) = \Theta(n^{2.807})$$

$$\boxed{T(n) = \Theta(n^{\lg 7})}$$

1.6 (f) $\mathbf{T(n) = 2T(n/4) + \sqrt{n}}$

$$a = 2, b = 4 \implies \beta = \log_4 2 = \log_2 / \log_4 = \log_2 / 2\log_2 = 0.5; \alpha = 0.5$$

Here, $\alpha > \beta$, thus:

$$T(n) = \Theta(n^\alpha \log n) = \Theta(n^{0.5} \log n) = \Theta(\sqrt{n} \log n)$$

$$\boxed{T(n) = \Theta(\sqrt{n} \log n)}$$

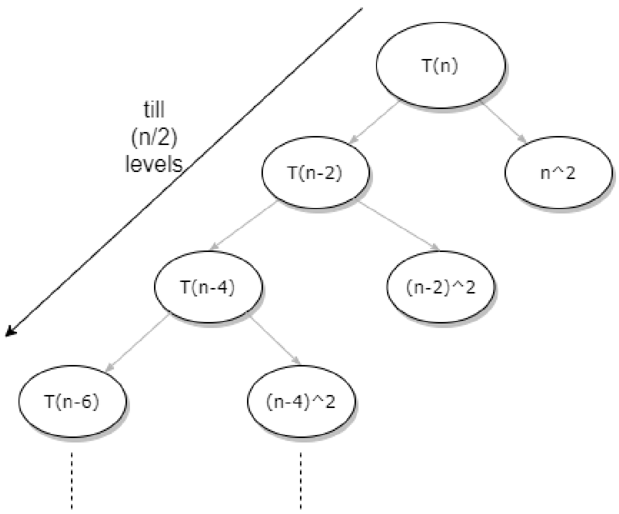

Figure 1: Recursive tree of $\frac{n}{2}$ levels for $T(n) = T(n-2) + n^2$

1.7 (g) $\mathbf{T(n) = T(n-2) + n^2}$

In Figure 1, we see the recursive tree for the given divide-conquer problem. Let's solve this one with recursive tree visualization approach.

From Figure 1, we can tell that the tree will have $n/2$ levels. Because, we are decreasing 2 elements on each level. E.g., if there is $n = 10$, there will be 5 levels where on first level: $n = 10$, on second level: $n = 8$, on third level: $n = 6$, on fourth level: $n = 4$ and on fifth level: $n = 2$. This is where the division stops and algorithm executes combine steps.

So, now we know that there will be $n/2$ levels and each level costs n^2.

$$\therefore \frac{n}{2} \cdot n^2 = \frac{n^3}{2}$$

Thus, we can say that $T(n) = T(n-2) + n^2 = \Theta(n^3)$.